Lesson 28: Web Scraping I

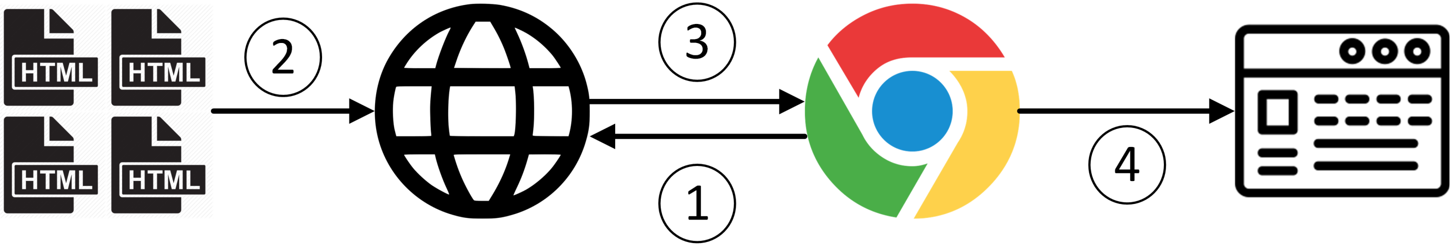
Ian Kloo

March 2022

Today

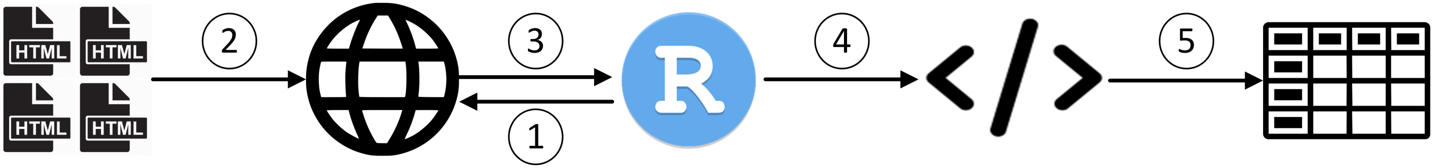
1. Understand basic principles of HTML
2. Understand basic principles of websites/web requests
3. Scrape single nodes from web pages

How the internet works



1) Request a website through a browser, with a URL
2) Internet routs your traffic to find the website (HTML file) to serve to you
3) HTML sent to your browser
4) HTML converted to what we see as a website

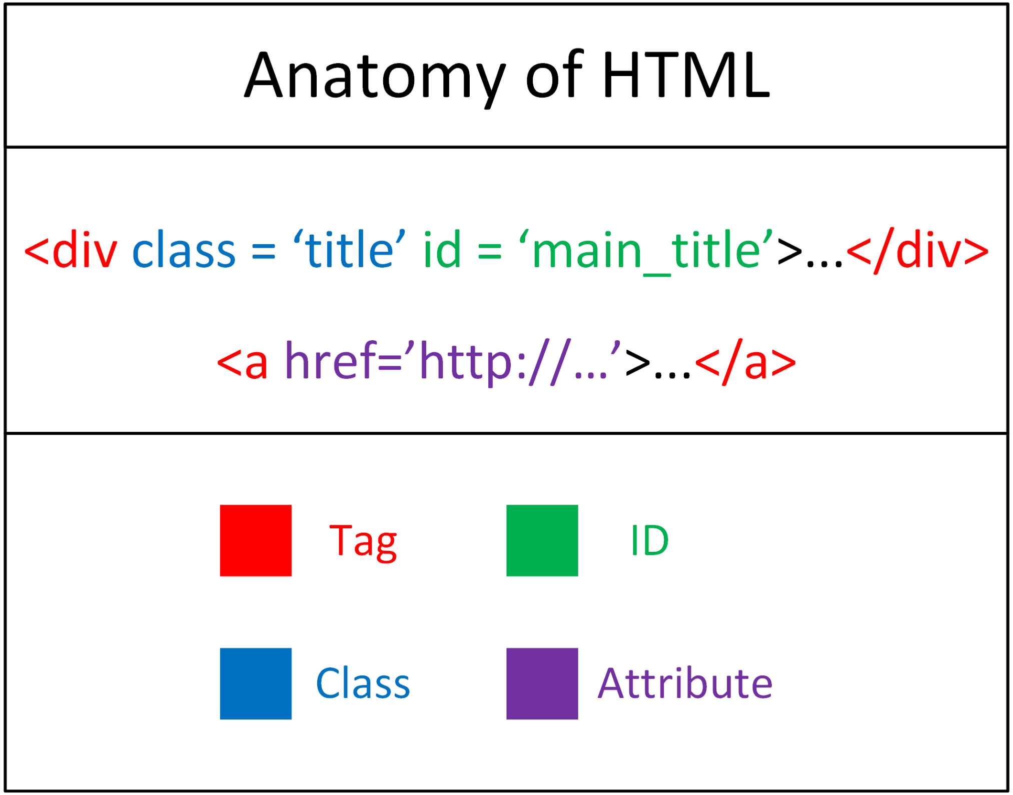
How web scraping works



1) Request a website through a R, with a URL
2) Internet routs your traffic to find the website (HTML file) to serve to you
3) HTML sent to R
4) Write code to extract data using HTML tags, classes, and ids
5) End up with data

HTML

- Let's talk about the basics of HTML



Anatomy of HTML

```
<div class = ‘title’ id = ‘main_title’>...</div>

<a href=’http://...’>...</a>
```

🟥 Tag 🟩 ID

🟦 Class 🟪 Attribute

HTML

- HTML is a markup language that is interpreted by a web browser
- Open notepad and write this text:

```
<h1>My Site Title</h1>

<h3>My Name</h3>

<p>My website is about...</p>

<a href = 'https://google.com'>Link to google</a>
```

- Open your file in a web browser...you made a website!

CSS

- CSS is a way to style HTML code
- Add this to the top of your file:

```
<style>
  h1{
    color: red;
  }
</style>
```

- Refresh your webpage to see how the styling is applied

CSS

- CSS relies on tags, classes, and ids to apply styling:
- Add some classes and apply the red font only to them:

```
<style>
  .my_color_class{
    color: red;
  }
</style>

<h1>My Site Title</h1>

<h3 class = 'my_color_class'>My Name</h3>

<p class = 'my_color_class'>My website is about...</p>

<a href = 'https://google.com'>Link to google</a>
```

- Note how you select a class with `.`
- You can select an id with #

CSS

- Let's add an id and color based on it:

```
<style>
  .my_color_class{
    color: red;
  }

  #my_id{
    color: blue;
  }

</style>

<h1 id = 'my_id'>My Site Title</h1>

<h3 class = 'my_color_class'>My Name</h3>

<p class = 'my_color_class'>My website is about...</p>

<a href = 'https://google.com'>Link to google</a>
```

Web Scraping

- For our purposes, we're not worried about web page style...
- But we can use the tags, classes, and ids to extract things from html!
- Check out: flukeout.github.io for practice with css selectors

Simple "web" scrape

- Let's scrape the file we just created!

```r
library(rvest)

read_html('test_site.html') %>%
  html_node('h1') %>%
  html_text()
```

```
## [1] "My Site Title"
```

- Or

```r
read_html('test_site.html') %>%
  html_node('#my_id') %>%
  html_text()
```

```
## [1] "My Site Title"
```

Simple "web" scrape

- We can also scrape classes:

```
read_html('test_site.html') %>%
  html_node('.my_color_class') %>%
  html_text()
```

```
## [1] "My Name"
```

- Or, if we want both of them

```
read_html('test_site.html') %>%
  html_nodes('.my_color_class') %>%
  html_text()
```

```
## [1] "My Name"                      "My website is about..."
```

Simple "web" scrape

- We can also scrape information from links:

```
read_html('test_site.html') %>%
  html_node('a') %>%
  html_text()
```

```
## [1] "Link to google"
```

```
read_html('test_site.html') %>%
  html_node('a') %>%
  html_attr('href')
```

```
## [1] "https://google.com"
```

"Real" Web Scraping

- Most web pages are more complicated than our simple one...
- ...but the process to scrape them is the same!
- The trick is finding the right selectors - so be patient and practice.

Legal Issues?

- Web scraping is not unethical or illegal if you are only accessing data you already have access to
  - For example, scraping ESPN headlines every 2 hours is functionally the same as manually copy/pasting them every 2 hours.

- However, scraping many websites is against their terms of use (though not illegal).

- It is very easy to accidentally spam a website with requests while scraping - this is why websites don't like it when people scrape their data.

- Be a good citizen and make sure make the fewest requests to a website that you can. This will ensure you don't get blocked.

- Rules to follow: Don't do anything via scraping you wouldn't do on a browser AND make sure you fully understand any web scraping code before you run it. Ignorance is not a valid excuse.